

Greibachové normální forma

Definice 3.33. Bezkontextová gramatika $\mathcal{G} = (N, \Sigma, P, S)$ je v **Greibachové normální formě** (GNF) právě když

- \mathcal{G} je bez ε -pravidel a
- každé pravidlo z P je tvaru $A \rightarrow a\alpha$, kde $a \in \Sigma$ a $\alpha \in N^*$ (s případnou výjimkou pravidla $S \rightarrow \varepsilon$).

Věta 3.34. Každý bezkontextový jazyk lze generovat bezkontextovou gramatikou v Greibachové normální formě.

Motivační příklad

$A \rightarrow BB \mid aAa \mid b$

$B \rightarrow bBa \mid bAb$

Lemma o substituci (pro připomenutí)

Lemma 3.20. (o substituci)

Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG. Nechť $A \rightarrow \alpha_1 B \alpha_2 \in P$.

Nechť $B \rightarrow \beta_1 \mid \dots \mid \beta_r$ jsou všechna pravidla v P tvaru $B \rightarrow \alpha$.

Definujme $\mathcal{G}' = (N, \Sigma, P', S)$, kde

$$P' = (P \setminus \{A \rightarrow \alpha_1 B \alpha_2\}) \cup \{A \rightarrow \alpha_1 \beta_1 \alpha_2 \mid \dots \mid \alpha_1 \beta_r \alpha_2\}.$$

Pak $L(\mathcal{G}) = L(\mathcal{G}')$.

Rekursivní neterminály a gramatiky

Definice 3.28. Neterminál A v CFG $\mathcal{G} = (N, \Sigma, P, S)$ se nazývá **levorekursivní** jestliže v \mathcal{G} existuje derivace $A \Rightarrow^+ A\beta$.

CFG bez levorekursivních neterminálů se nazývá **nelevorekursivní**.

Převod bezkontextové gramatiky \mathcal{G} do GNF:

$L(\mathcal{G})$ je prázdný: zřejmé ($S \rightarrow aS$)

$L(\mathcal{G})$ je neprázdný: 1. z vlastní gramatiky eliminujeme levou rekurzi
2. pak převedeme do GNF

Algoritmus odstranění přímé levé rekurze

Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je bezkontextová gramatika bez jednoduchých a ε -pravidel, v níž všechna A -pravidla (pravidla mající na levé straně A) jsou tvaru

$$A \rightarrow A\alpha_1 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \dots \mid \beta_n,$$

kde každý řetěz β_i začíná symbolem různým od A .

Nechť $\mathcal{G}' = (N \cup \{A'\}, \Sigma, P', S)$, kde P' obdržíme z P tak, že všechna výše uvedená pravidla nahradíme pravidly

$$A \rightarrow \beta_1 \mid \dots \mid \beta_n \mid \beta_1 A' \mid \dots \mid \beta_n A'$$

$$A' \rightarrow \alpha_1 \mid \dots \mid \alpha_m \mid \alpha_1 A' \mid \dots \mid \alpha_m A'$$

Pak $L(\mathcal{G}) = L(\mathcal{G}')$.

Příklad

$A \rightarrow Bd \mid c$

$B \rightarrow Bdd \mid Ccc \mid aAd$

$C \rightarrow Aa$

Algoritmus odstranění levé rekurze

Vstup: Vlastní CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: Ekvivalentní nelevorekursivní gramatika

```
1 Uspořádej libovolně  $N$ ,  $N = \{A_1, \dots, A_n\}$ 
2 for  $i \leftarrow 1$  to  $n$  do
3   for  $j \leftarrow 1$  to  $i - 1$  do
4     foreach pravidlo tvaru  $A_i \rightarrow A_j\alpha$  do
5       přidej pravidla  $A_i \rightarrow \beta_1\alpha \mid \dots \mid \beta_k\alpha$ 
6       (kde  $A_j \rightarrow \beta_1 \mid \dots \mid \beta_k$  jsou všechna  $A_j$ -pravidla)
7       vypusť pravidlo  $A_i \rightarrow A_j\alpha$ 
8   od
9 od
10   odstraň případnou přímou levou rekurzi na  $A_i$ 
11 od
```

Korektnost algoritmu

Konečnost.

Ekvivalence gramatik: Všechny úpravy jsou dle Lemmatu o substituci nebo odstraňují přímou levou rekurzi.

Výsledná gramatika je nelevorekursivní:

1. po i -té iteraci vnějšího cyklu začíná každé A_i -pravidlo buď terminálem nebo neterminálem A_k , kde $k > i$.
2. po j -té iteraci vnitřního cyklu začíná každé A_i -pravidlo buď terminálem nebo neterminálem A_k , kde $k > j$.

Příklad

$$A \rightarrow Ba \mid Db \mid c$$
$$B \rightarrow CC$$
$$C \rightarrow aE$$
$$D \rightarrow CDa \mid Eb$$
$$E \rightarrow bb$$

Algoritmus transformace do GNF

Vstup: Nelevorekursivní CFG $\mathcal{G} = (N, \Sigma, P, S)$ bez ε -pravidel

Výstup: CFG \mathcal{G}' v GNF splňující $L(\mathcal{G}) = L(\mathcal{G}')$

- 1 najdi lineární uspořádání \prec splňující $(A \rightarrow B\alpha) \in P \implies A \prec B$
- 2 Označme $N = \{A_1, \dots, A_n \mid A_{i-1} \prec A_i, 1 < i \leq n\}$
- 3 **for** $i \leftarrow n - 1$ **downto** 1 **do**
- 4 **foreach** pravidlo tvaru $A_i \rightarrow A_j\alpha$, kde $j > i$ **do**
- 5 přidej pravidlo $A_i \rightarrow \beta_1\alpha \mid \dots \mid \beta_k\alpha$
- 6 (kde $A_j \rightarrow \beta_1 \mid \dots \mid \beta_k$ jsou všechna A_j -pravidla)
- 7 vypuť pravidlo $A_i \rightarrow A_j\alpha$
- 8 **od**
- 9 **od**
- 10 nahraď potřebné terminály novými neterminály
- 11 a přidej příslušná pravidla

Korektnost algoritmu

Konečnost.

Ekvivalence gramatik.

Výsledná gramatika je v GNF.

Zásobníkové automaty

Definice zásobníkového automatu

Definice 3.36. Nedeterministický zásobníkový automat (PushDown Automaton, PDA) je sedmice $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- Q je konečná množina, jejíž prvky nazýváme **stavy**,
- Σ je konečná množina, tzv. **vstupní abeceda**,
- Γ je konečná množina, tzv. **zásobníková abeceda**,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \mathcal{P}_{Fin}(Q \times \Gamma^*)$, tzv. (parciální) **přechodová funkce**¹,
- $q_0 \in Q$ je **počáteční stav**,
- $Z_0 \in \Gamma$ je **počáteční symbol v zásobníku**,
- $F \subseteq Q$ je množina **koncových stavů**.

¹Zápis $\mathcal{P}_{Fin}(Q \times \Gamma^*)$ značí množinu všech **konečných** podmnožin množiny $Q \times \Gamma^*$.

Výpočet zásobníkového automatu

Definice 3.37. Nechť $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je PDA.

Konfigurací nazveme libovolný prvek $(p, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$.

Na množině všech konfigurací automatu \mathcal{M} definujeme binární relaci **krok výpočtu** $\vdash_{\mathcal{M}}$ takto:

$$(p, aw, Z\alpha) \vdash_{\mathcal{M}} (q, w, \gamma\alpha) \stackrel{def}{\iff} \exists (q, \gamma) \in \delta(p, a, Z) \text{ pro } a \in \Sigma \cup \{\varepsilon\}$$

Reflexivní a tranzitivní uzávěr relace $\vdash_{\mathcal{M}}$ značíme $\vdash_{\mathcal{M}}^*$.

Je-li \mathcal{M} zřejmý z kontextu, píšeme pouze \vdash resp. \vdash^* .

Akceptující výpočet zásobníkového automatu

Definice 3.37.(pokračování)

Jazyk akceptovaný PDA \mathcal{M} koncovým stavem definujeme jako

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \xrightarrow{*} (q_f, \varepsilon, \alpha), \text{ kde } q_f \in F, \alpha \in \Gamma^*\}$$

a jazyk akceptovaný PDA \mathcal{M} **prázdným zásobníkem** definujeme

$$L_e(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \xrightarrow{*} (q, \varepsilon, \varepsilon), \text{ kde } q \in Q\}.$$